MCTS Overview 687

Sean Folan

November 2024

1 Overview

Monte Carlo Tree Search (MCTS) is an online algorithm that attempts to estimate the q function of a state though random simulation. Everytime it visits a state, it builds an ExpectiMax search tree incrementally. The search can be terminated after a given amount of time or an amount of expanded nodes. Typically more useful for huge state spaces, famously used in AlphaGo (Go) and Pluribus (No limit Texas Hold'em Poker).

2 Algorithm

The Algorithm has four parts, which are repeated until the computational budget is met.

1. **Selection** - Select an unexpanded node.
2. **Expansion** - If we are not in a terminal state, we expand one or more of the children nodes
3. **Simulation** - Choose one of the new nodes and perform Monte Carlo simulation of the MDP
4. **Backpropagation** - The return is backpropagated up to the root

Steps one and two are defined by a TreePolicy which tells the algorithm how to select and expand and step three utilizes DefaultPolicy which encodes how the simulations are carried out.

2.1 Upper Confidence Trees

One of the most popular tree policies is Upper Confidence Trees (UCT). This strategy has us pick nodes to...

$$\arg\max_{a \in A} Q(s,a) + 2C_p \sqrt{\frac{2 \ln N(s)}{N(s,a)}}$$

3 Pseudocode

Algorithm 1 MCTS

Input: MDP $M = (\mathcal{S}, \mathcal{A}, p, d_0, R, \gamma)$, Time limit T, current state s_0
Output: Estimated Q function
while $time < T$ **do**
    node $\leftarrow$ Select(s_0) ▷ Find a node that is not fully explored
    child $\leftarrow$ Expand(node) ▷ Expand the node to get the node you will start the Simulation from
    $G \leftarrow$ Simulate(child) ▷ Run the episode getting return G
    Backpropagate(node, G) ▷ Return results all the way up to the parent node
end while

4 Sources

1. https://gibberblot.github.io/rl-notes/single-agent/mcts.html
2. http://incompleteideas.net/book/RLbook2020.pdf
3. http://www.incompleteideas.net/609%20dropbox/other%20readings%20and%20resources/MCTS-survey.pdf
4. https://courses.cs.washington.edu/courses/cse473/11au/slides/cse473au11-adversarial-search.pdf